EE219 Large-scale Data Mining

Course Project #5

Popularity Prediction on Twitter

Shijun Lu（905035448）

Wenfei Lu（505035450）

Xingyi Chen（205032924）

Hanren Lin（304944990）

Introduction

The prediction for a future event is becoming a critical part in social network analysis. In this project, we used Twitter, which is one of the most popular social interactive platforms, to carry out prediction and analysis based on reinforcement learning.

The available Twitter data is collected by querying popular hashtags related to the 2015 Super Bowl spanning a period starting from 2 weeks before the game to a week after the game. We used data from some related hashtags to train a regression model, and then use the model to make predictions on popularity and fan base for other hashtags.

Part 1: Popularity Prediction

<Problem 1.1>

For this problem, we downloaded the training tweet data and calculated the three statistics for each hashtag: average number of tweets per hour, average number of followers of users posting the tweets, and average number of retweets. The results are shown in Form 1.

Hashtags	Total number of tweets	Average number of tweets per hour	Average number of followers of users posting the tweets	Average number of retweets
#GoHawks	188136	325.37	2203.93	2.015
#GoPatriots	26232	45.69	1401.90	1.400
#NFL	259024	441.32	4653.25	1.539
#Patriots	489713	834.56	3309.98	1.783
#SB49	826951	1419.89	10267.32	2.511
#SuperBowl	1348767	2302.50	8858.97	2.388

Form 1 – Several statistic properties for each hashtag

Each tweet file from the dataset contains one tweet in each line and tweets are sorted with respect to their posting time. Each tweet is a JSON string, which can be loaded in Python as a dictionary. Therefore, we could quote the time, the number of retweets, the number of followers of the person retweeting from each tweet file with appropriate code applied.

The plotted result of number of tweets versus time for two hashtags #NFL and #SuperBowl are separately shown in Figure 1, 2.

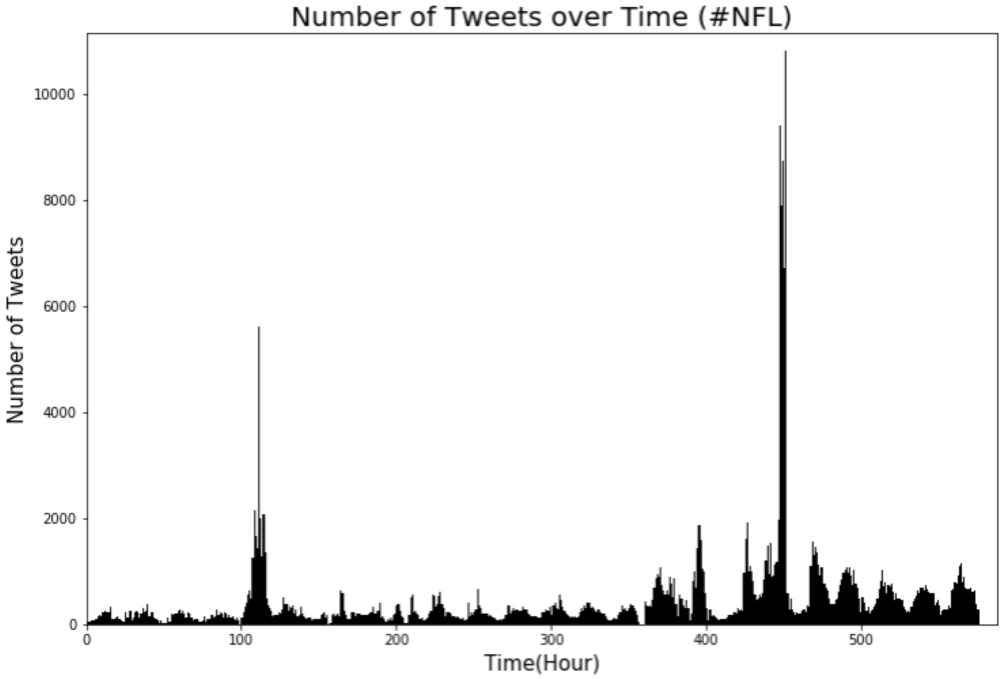

Figure 1 - Number of Tweets versus time (#NFL)

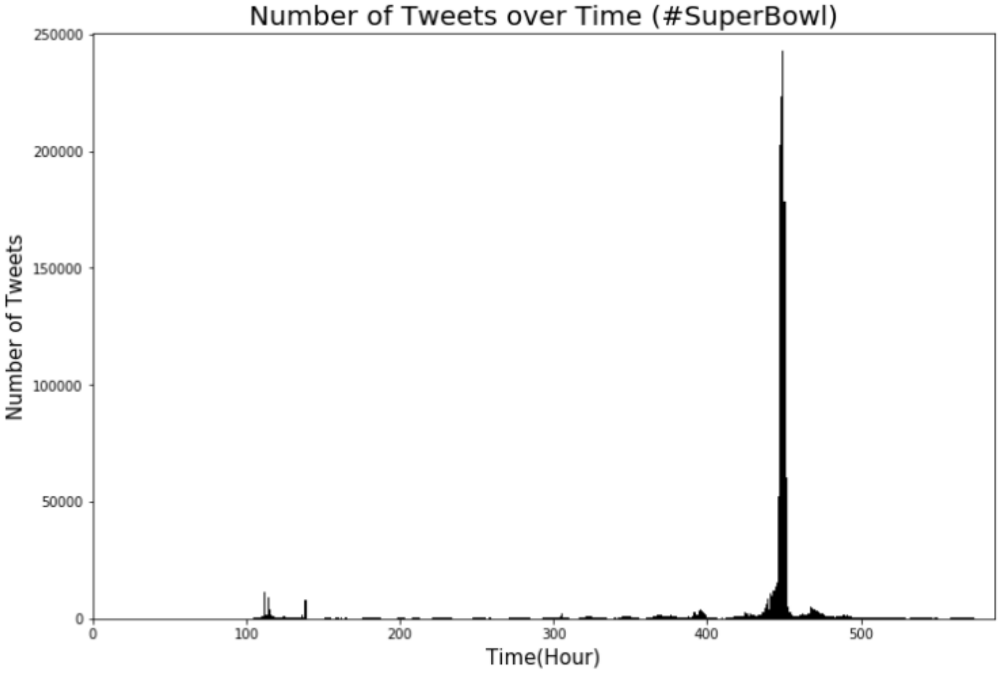

Figure 2 - Number of Tweets versus time (#SuperBowl)

<Problem 1.2>

In this problem, we fitted a Linear Regression model by applying 5 features for each hashtag for the prediction of tweets in the next hour, with the features extracted from tweet data in the previous hour. Each hashtag corresponds to a model that needs to be trained. The applied 5 features are:

- Number of tweets (hashtag of interest)
- Total number of retweets (hashtag of interest)
- Sum of the number of followers of the users posting the hashtag
- Maximum number of followers of the users posting the hashtag
- Time of the day (which could take 24 values that represent hours of the day with respect to a given time zone)

For the feature extraction, we used a time window, which could provide a divided sample with the same period for the regression model. In that case, we could use one certain hour's tweet data features to predict the next hour's number of tweets. To analyze the significance of each feature, t-test and P-value are introduced in this project.
The training accuracy (evaluated by RMSE), R-squared measure for each hashtag's training model are shown in Form 2, with T-test values and P-values for each feature separately reported in Form 3, 4, and the plotted results for relative error versus actual values for each hashtag are shown in Figure 3-14.

Hashtag	Training Accuracy (RMSE)	R-squared Measure
#GoHawks	938.8329	0.527
#GoPatriots	193.6321	0.610
#NFL	588.7724	0.639
#Patriots	2356.5450	0.717
#SB49	3875.3608	0.852
#SuperBowl	6600.9224	0.864

Form 2 – Training accuracy and R-squared measure for each hashtag

<T-test>	Num_tweets	Num_retweets	Sum_followers	Max_followers	Time_of_the_day
#GoHawks	9.127	-5.376	-3.659	1.555	2.306
#GoPatriots	1.262	-2.901	3.227	-3.519	1.749
#NFL	5.148	-0.443	2.328	-1.620	3.451
#Patriots	21.572	-5.623	1.321	1.701	0.857
#SB49	32.322	-7.256	0.756	4.496	-0.854
#SuperBowl	27.438	-2.204	-19.431	10.487	-2.285

Form 3 – T-test value for 5 features from each hashtag

<P-value>	Num_tweets	Num_retweets	Sum_followers	Max_followers	Time_of_the_day
#GoHawks	0.000	0.000	0.000	0.120	0.021
#GoPatriots	0.207	0.004	0.001	0.000	0.081
#NFL	0.000	0.658	0.020	0.106	0.001
#Patriots	0.000	0.000	0.187	0.090	0.392
#SB49	0.000	0.000	0.450	0.000	0.393
#SuperBowl	0.000	0.028	0.000	0.000	0.023

Form 4 – P-value for 5 features from each hashtag

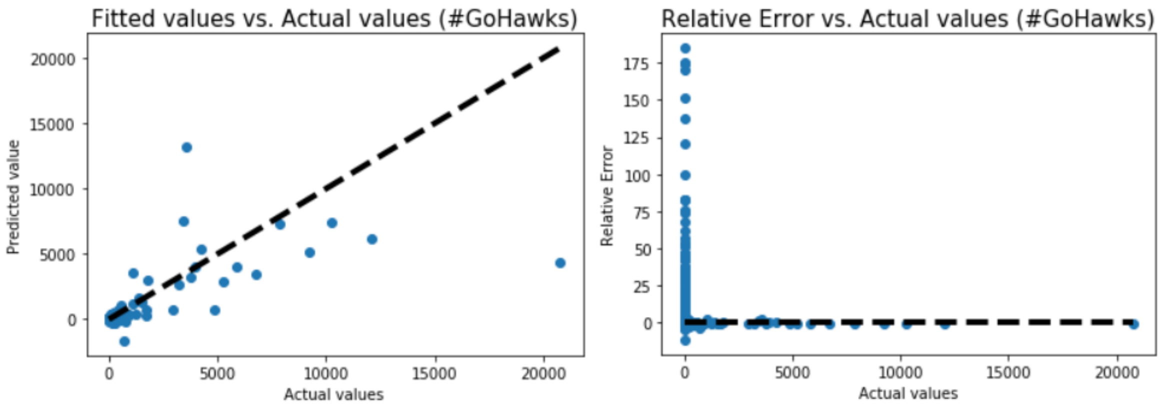

Figure 3 & 4 – Plotted result for fitted values / relative error versus actual values (#GoHawks)

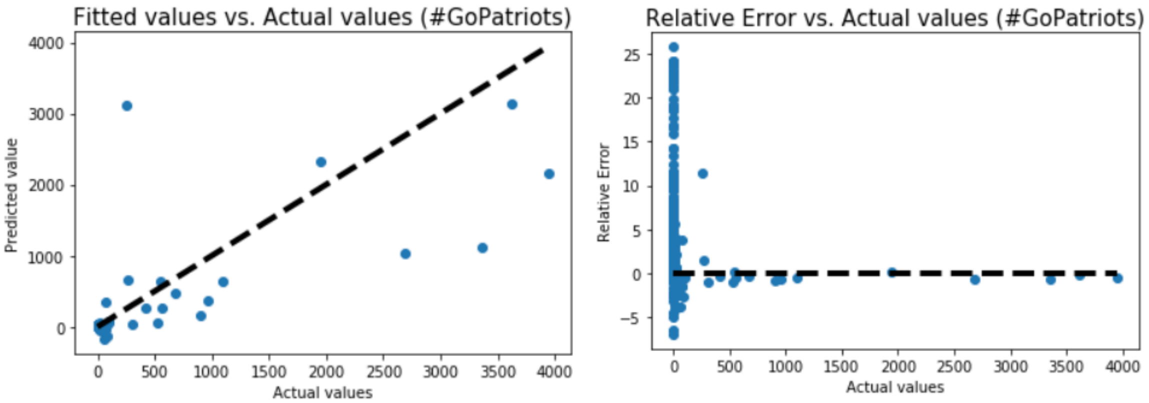

Figure 5 & 6 – Plotted result for fitted values / relative error versus actual values (#GoPatriots)

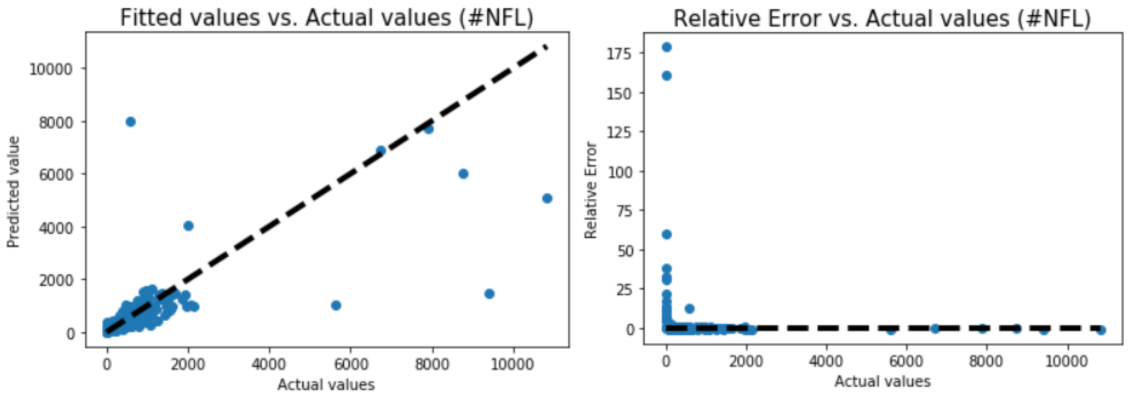

Figure 7 & 8 – Plotted result for fitted values / relative error versus actual values (#NFL)

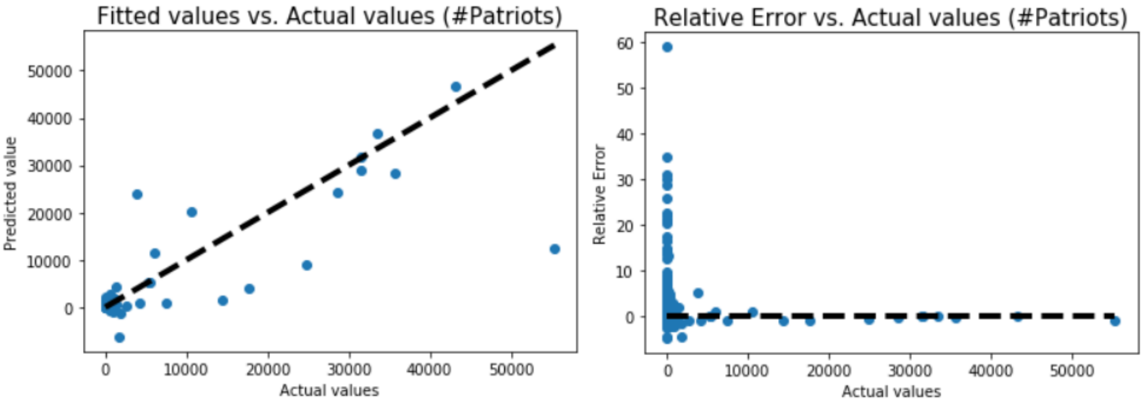

Figure 9 & 10 – Plotted result for fitted values / relative error versus actual values (#Patriots)

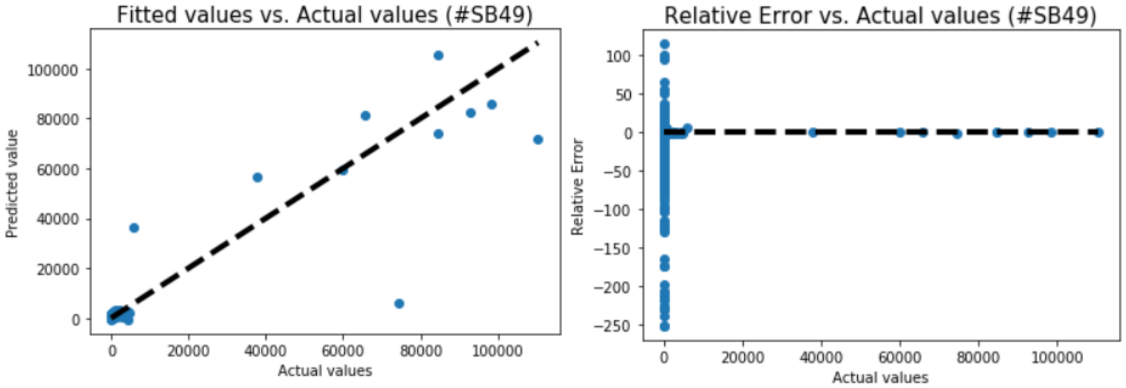

Figure 11 & 12 – Plotted result for fitted values / relative error versus actual values (#SB49)

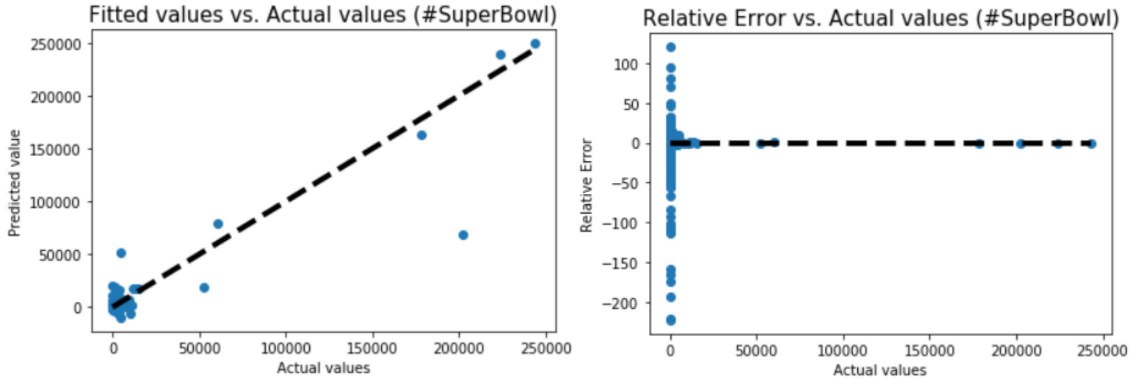

Figure 13 & 14 – Plotted result for fitted values / relative error versus actual values (#SuperBowl)

T-test is a statistic parameter that evaluates how significantly this feature would change the difference between two datasets, indicating that a high t-test value would define this feature as the "significant" one. P-value estimates the robustness of the feature against the "null hypothesis", and when P-value gets smaller (usually less than 0.05), the influence of "null hypothesis" would be smaller, proving the "strong" evidence and a feature with more significance.

From the t-test results shown above, each category has different features with the most significance. For instance, in the data from hashtag #NFL, the number of tweets, time of the day and sum of the followers both have a low P-value (less than 0.02), and the absolute value of t-test value are relatively higher among all 5 features. In contrast, the other 2 features have a P-value over 0.1, indicating the poor robustness against the “null hypothesis” and less significance to the dataset. With a relatively larger absolute value of t-test value and smaller P-value, the top 3 features for each hashtag are shown in Form 5.

Hashtag	Top 3 features
#GoHawks	Num_tweets, num_retweets, sum_followers
#GoPatriots	Max_followers, sum_followers, num_retweets
#NFL	Num_tweets, time_of_the_day, sum_followers
#Patriots	Num_tweets, num_retweets, max_followers
#SB49	Num_tweets, num_retweets, max_followers
#SuperBowl	Num_tweets, sum_followers, max_followers

Form 5 – Top 3 features for each hashtag (Based on t-test & P-value)

<Problem 1.3>

In this problem, we selected our own features for the regression model. The new features should reflect the “identity” of the dataset within a certain period. Based on this principle, we chose number of URLs (num_URLs), number of authors (num_authors), number of replies (num_replies), number of impressions (num_impressions), number of favorites (num_favorites), ranking score (ranking_score) and number of hashtags (num_hashtags) as the new 7 features in this problem. For instance, the number of URLs reflects the number of tweets indirectly, and number of hashtags indicates the hot spot of topics within certain period. For these new features, their fitting accuracy and significance of variables are reported. For a more intuitive evaluation, the t-test an p-value results are shown in Form 6, 7.

<T-test>	Num_URLs	Num_authors	Num_impressions	Ranking_score	Num_hashtags	Num_replies	Num_favorites
#GoHawks	8.324	4.713	1.516	14.496	-0.765	-1.203	0.773
#GoPatriots	14.749	-5.855	-9.475	9.196	15.739	-10.521	1.212
#NFL	0.320	-6.869	0.737	1.313	10.147	0.056	-6.706
#Patriots	11.270	-0.735	-2.246	9.837	5.749	-2.135	0.903
#SB49	13.672	-0.208	-1.754	9.985	3.859	-7.135	-2.276
#SuperBowl	3.472	14.192	-2.531	9.991	2.582	-5.675	-10.938

Form 6 – T-test results for other 5 features

<P-Value>	Num_URLs	Num_authors	Num_impressions	Ranking_score	Num_hashtags	Num_replies	Num_favorites
#GoHawks	0.000	0.000	0.130	0.000	0.445	0.229	0.440
#GoPatriots	0.000	0.000	0.000	0.000	0.000	0.000	0.226
#NFL	0.749	0.000	0.461	0.190	0.000	0.955	0.000
#Patriots	0.000	0.463	0.025	0.000	0.000	0.033	0.367
#SB49	0.000	0.835	0.080	0.000	0.000	0.000	0.023
#SuperBowl	0.001	0.000	0.012	0.000	0.010	0.000	0.000

Form 7 – P-value results for other 5 features

Then we should select the top 3 valuable features which perform relatively linear relationship between features we selected and target value. The 3 features with most significance we found for each hashtag in this section are shown in Form 8, which are selected based on a higher t-test value and a lower P-value as demonstrated above.

Hashtag	Top 3 significant Features
#GoHawks	Ranking_score, num_tweets, num_URLs
#GoPatriots	Num_hashtags, num_URLs, num_replies
#NFL	Num_hashtags, num_authors, num_favorites
#Patriots	Num_URLs, ranking_score, num_tweets
#SB49	Num_URLs, num_tweets, ranking_score
#SuperBowl	Num_authors, num_tweets, num_favorites

Form 8 – Selected 3 features from each hashtag

A scatter figure with number of tweets for next hour versus value of the selected feature was plotted for each of the top 3 features. The plotted results for each hashtag are separately shown in Figure 15-32.

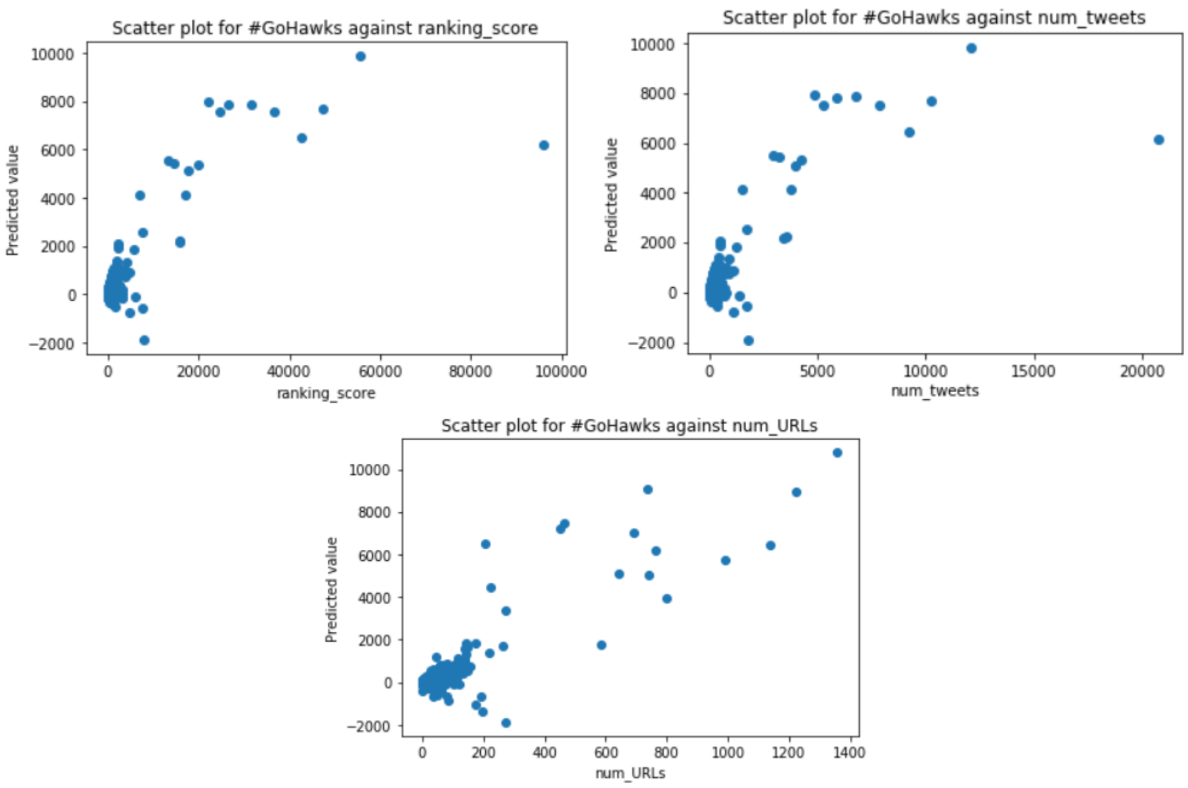

Figure 15-17 – Top 3 Features' predicted value scatter plot (#GoHawks)

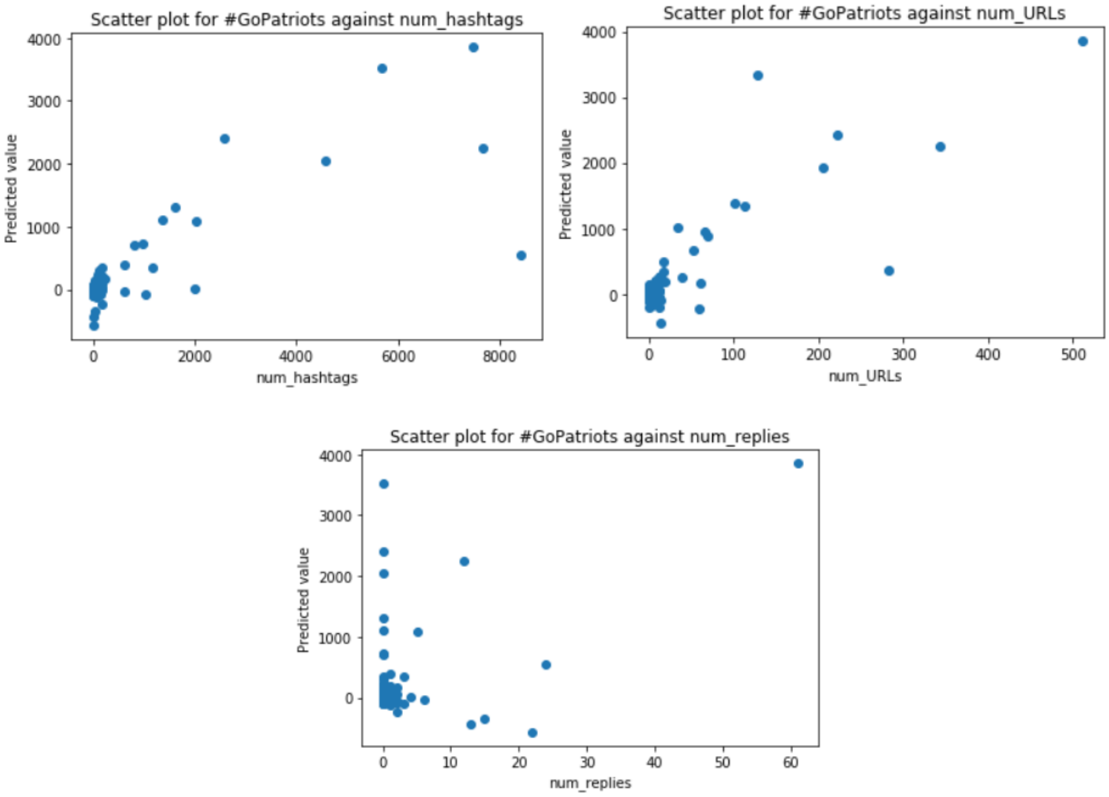

Figure 18-20 – Top 3 Features' predicted value scatter plot (#GoPatriots)

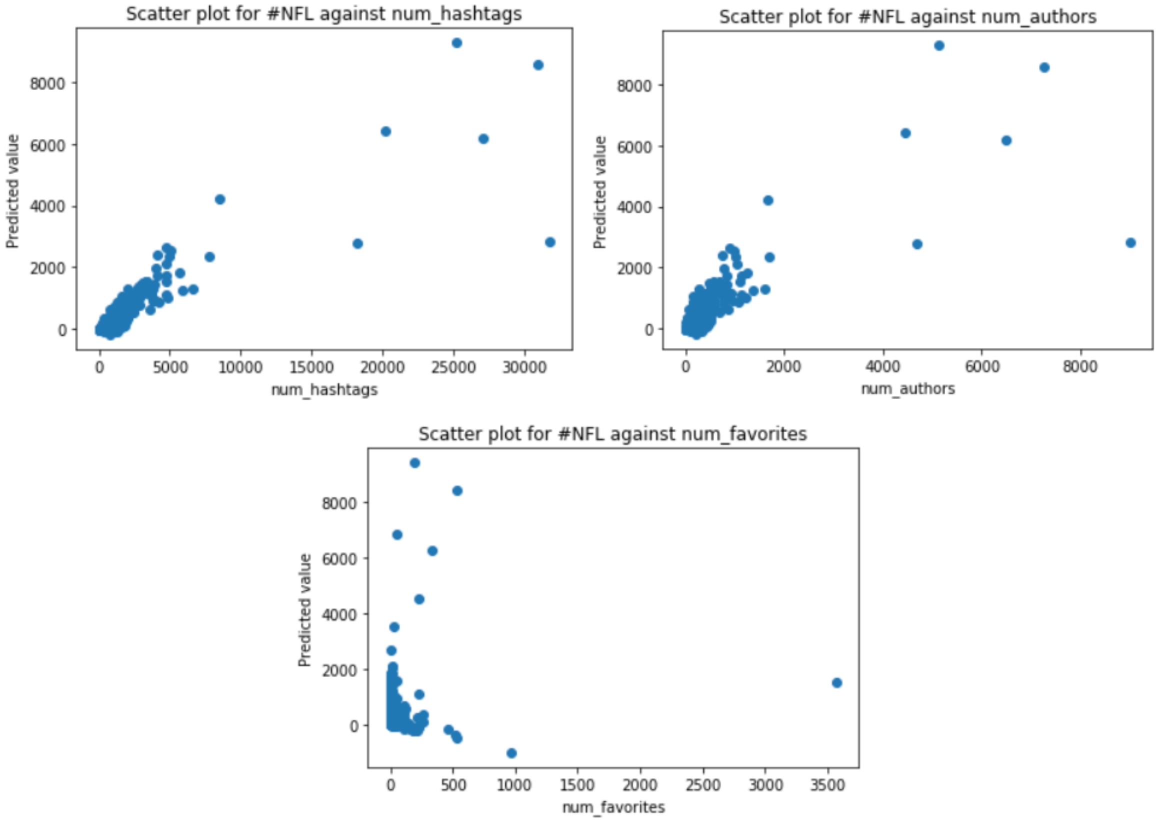

Figure 21-23 – Top 3 Features' predicted value scatter plot (#NFL)

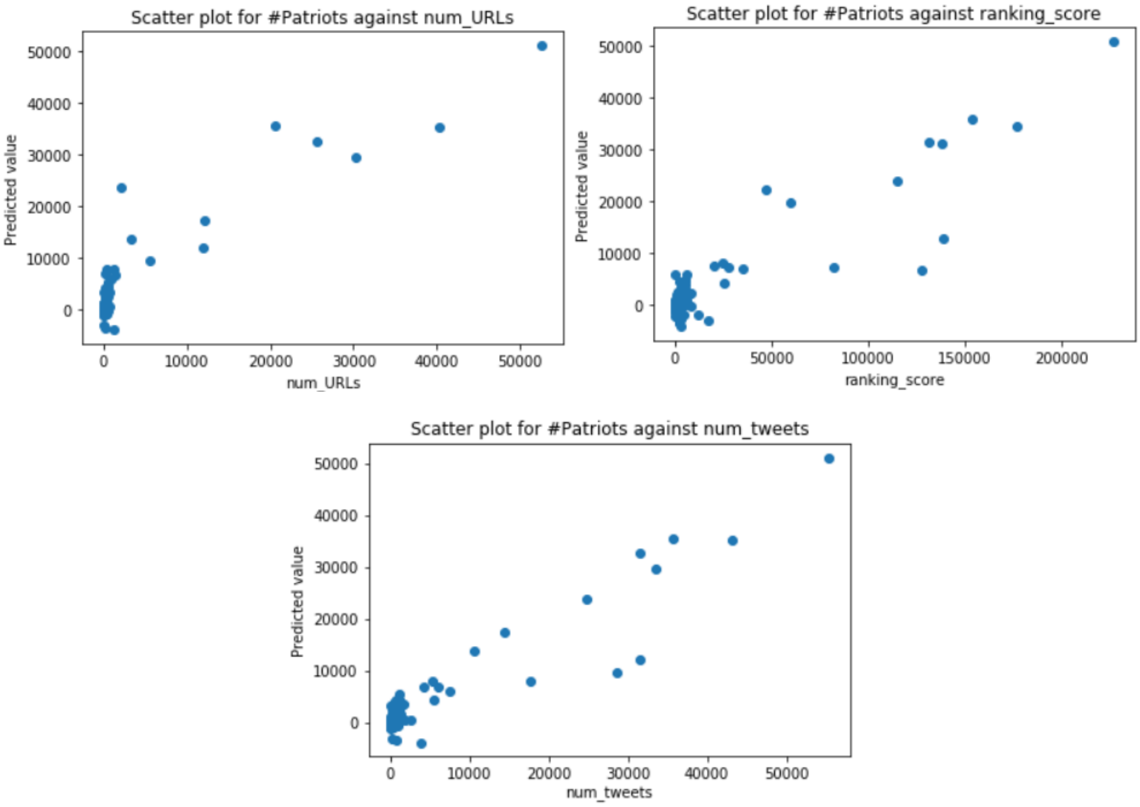

Figure 24-26 – Top 3 Features' predicted value scatter plot (#Patriots)

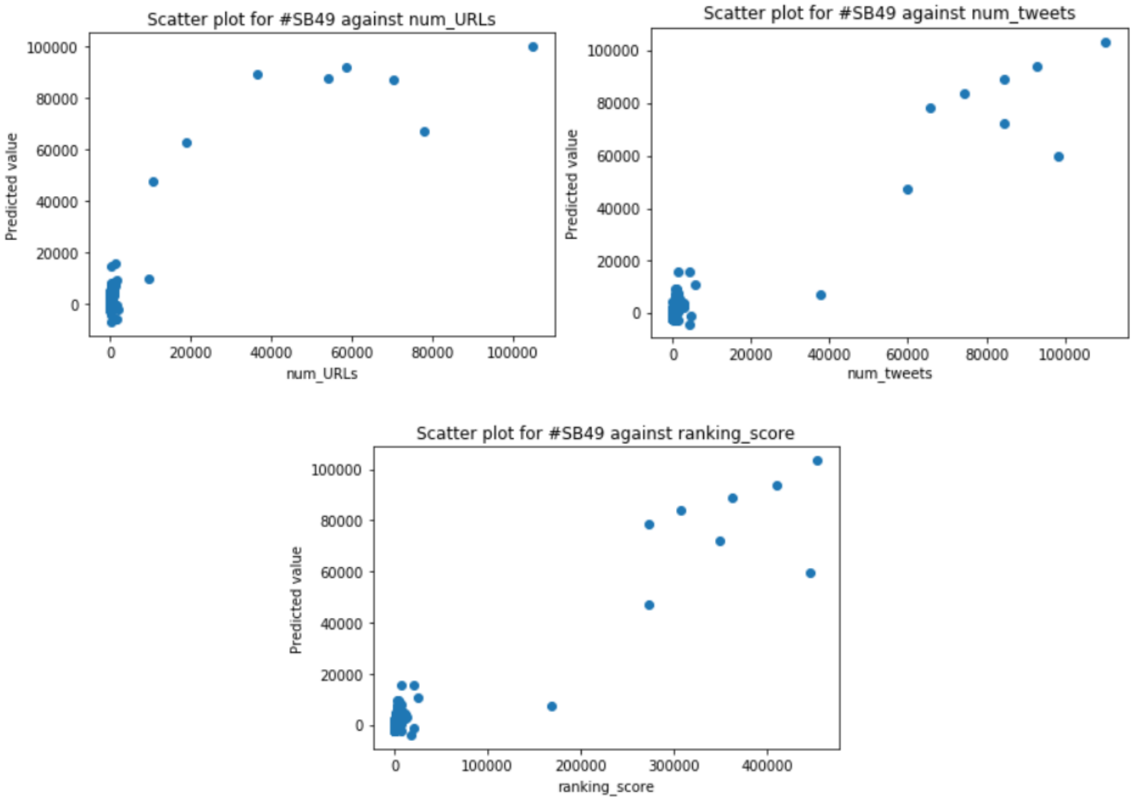

Figure 27-29 – Top 3 Features' predicted value scatter plot (#SB49)

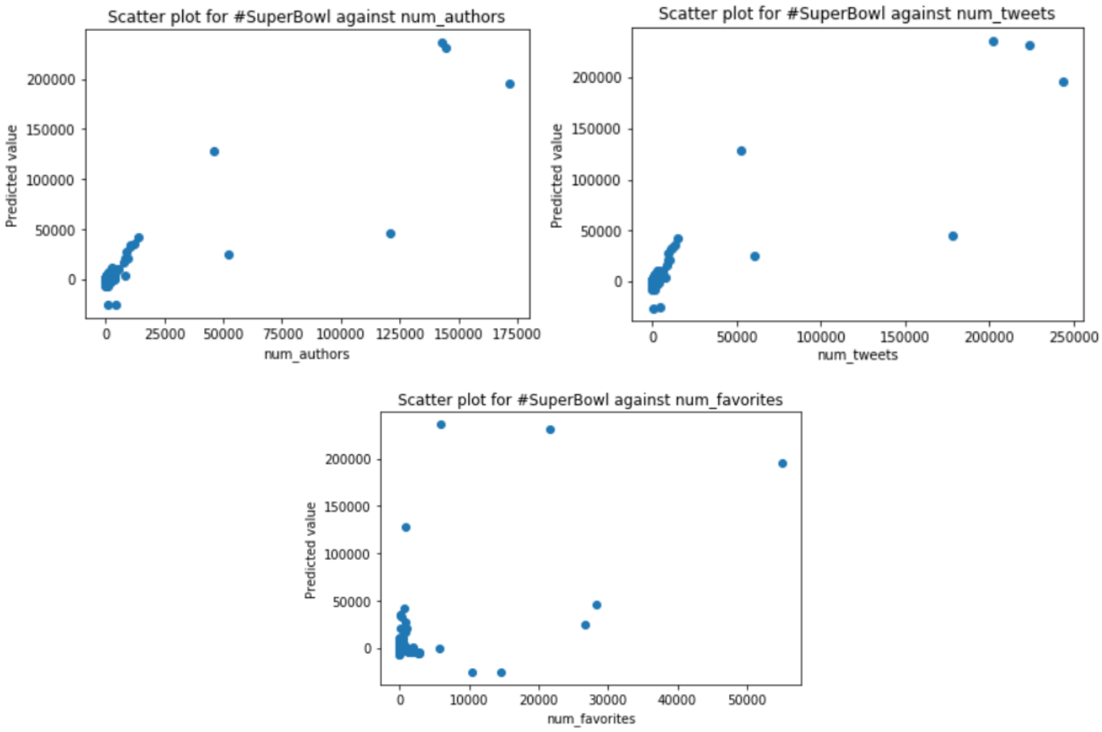

Figure 30-32 – Top 3 Features' predicted value scatter plot (#SuperBowl)

From the plotted results, the predicted value tends to concentrate on bottom left of each figure. Plus, they also show the linear relationship of different hashtags. For instance, the relationship between features and predicted values of #NFL is more linear than other hashtags, which indicates that we may have less error in predicting.

<Problem 1.4 Q1>

In this problem, we are supposed to apply k-fold cross validation to the models introduced in the previous parts, with average prediction (cross-validation) error reported for each hashtag in the evaluation.

To proceed the k-fold cross validation, we need to divide the training data into k parts, and then execute k tests, with k-1 parts being the dataset used for each fitting process. In this section, we used k=10, and the evaluated error can be reported in the form of $|N_{predicted} - N_{real}|$.

We built 3 different regression models for 3 different time periods with various extent of activity:

- Before Feb. 1, 8:00 a.m.
- Between Feb. 1, 8:00 a.m. and 8:00 p.m.
- After Feb. 1, 8:00 p.m.

The three types of models include one linear regression model and two non-linear regression models, which are random forest regression and neural network regression.

We set the start time at 2015-01-14-00:00:00 and divided the time for each time period. Then we calculated the cross-validation error for each time period and repeated this process for all hashtags.

The average cross-validation errors for the 3 different types of models with separated data are reported in Form 9-11.

Hashtags\Models	Before Feb. 1, 8:00 a.m.	Between Feb. 1, 8:00 a.m. and 8:00 p.m.	After Feb. 1, 8:00 p.m.
#GoHawks	411.3435	4440.7096	50.2120
#GoPatriots	19.9038	5442.6622	4.7688
#NFL	126.9146	5450.4256	152.0705
#Patriots	305.1296	66679.8467	158.0314
#SB49	58.6017	343451.5215	158.5759
#SuperBowl	396.1154	387389.1938	643.7562

Form 9 – Average prediction error with separated data (Linear Regression)

Hashtags\Models	Before Feb. 1, 8:00 a.m.	Between Feb. 1, 8:00 a.m. and 8:00 p.m.	After Feb. 1, 8:00 p.m.
#GoHawks	154.7817	1993.5417	25.0440
#GoPatriots	11.1871	1242.2208	3.3802
#NFL	108.2843	2804.4792	147.7735
#Patriots	204.5978	17056.775	103.6506
#SB49	47.2834	24011.9500	107.0465
#SuperBowl	264.2315	65071.9625	331.8385

Form 10 – Average prediction error with separated data (Random Forest Regression)

Hashtags\Models	Before Feb. 1, 8:00 a.m.	Between Feb. 1, 8:00 a.m. and 8:00 p.m.	After Feb. 1, 8:00 p.m.
#GoHawks	20995.3962	158183.0783	2779.6283
#GoPatriots	319.4402	35746.2637	251.8618
#NFL	48122.5456	285588.6629	42731.5898
#Patriots	53268.5605	1128804.5458	29500.7729
#SB49	80025.2874	9842218.7301	298386.8889
#SuperBowl	149207.7260	13329257.7380	389007.7728

Form 11 – Average prediction error with separated data (Neural Network Regression)

From the results shown above, when applying random forest regression model, we could get a minimized average prediction error. Therefore, for the fitting process with aggregated data, we also used the same model for evaluation.

<Problem 1.4 Q2>

In this problem, we use random forest regression model for prediction. First, we generated a file with all data from each hashtag and then calculate the average cross-validation error.

The average prediction errors with aggregated data are shown in Form 12.

Models	Error with Aggregated Data
Before Feb. 1, 8:00 a.m.	651.8355
Between Feb. 1, 8:00 a.m. and 8:00 p.m.	113057.3292
After Feb. 1, 8:00 p.m.	394.0635

Form 12 – Average prediction error with aggregated data (Random forest regression)

The data from each hashtag would tend to have more relevance with the data within the same hashtag, so making the prediction from the same hashtag's dataset would have a more precise result. From the average prediction results above, when using separated data for the evaluation, we could get a smaller average prediction error result, indicating the model having a better performance.

Plus, we can see that the error of the second time period is much more larger than the others. That is because the length of time is different. The second time period only contains 12 hours of data, but the first and the last time period contains much more time. And from the plot of Problem 1.1, the number of tweets of the second period is much more larger and random than the others. Moreover, as we can see from the plot of Problem 1.1, the first time period has a sudden increase of tweets at around 100—120 hours. Hence, for the first and last time period, although the first time period contains larger time range, it's more random than the last time period and has a larger error.

In addition, appropriately applying non-linear regression model would improve the overall performance.

<Problem 1.5>

A new dataset with each hashtag's tweets having a 6-hour window was used for this section. For this problem, we are supposed to predict the number of tweets in the next hour, which is right after the 6-hour window, from the data in the whole 6-hour window rather than that in the previous one hour. To select a better model, we used the training data to fit several models, with using cross-validation as the evaluation method. The models we used are linear regression model and random forest model. Under two models, the average prediction errors in different periods are shown in Form 13.

	Linear Regression	Random Forest
Before Feb. 1, 8:00 a.m.	5470.9876	1922.0067
Between Feb. 1, 8:00 a.m. and 8:00 p.m.	81748.2162	87542.4708
After Feb. 1, 8:00 p.m.	757.5310	656.2495

Form 13 – Average prediction errors in different models

From Form 13, the random forest regression model provides a relatively satisfactory prediction, therefore we still apply this model in this problem.
The results of predicting 10 file's number of tweets are shown in Form 14, 15.

File Name	Sample1_period1.txt	Sample2_period2.txt	Sample3_period3.txt	Sample4_period1.txt	Sample5_period1.txt
Prediction	475.95	4485.55	539.125	472.7	493.55

File Name	Sample6_period2.txt	Sample7_period3.txt	Sample8_period1.txt	Sample9_period2.txt	Sample10_period3.txt
Prediction	4728.4	36.8	66.95	1689.965	53.65

Form 14 & 15 – Prediction Result of next hour's number of tweets

The prediction results have some obvious patterns. When predicting from the data in period 2, the prediction result would be much higher than predicting from the data in period 1 and 3. A possible reason is that within the period a few hours before the Super Bowl Night or in the game, the number of tweets on relevant topics would appear a significant peak.

Part 2: Fan Base Prediction

In this part, we are supposed to predict the location of the author of a tweet.

A tweet could directly or indirectly reveal some information about the tweet's author. For instance, the author's location would possibly indicate the author's favorite team. For the problem purpose, we analyzed the data from tweets including #superbowl, and predicted the location of the author via the self-trained binary classifier.

The classification algorithms we applied in this part are: SVM classifier, Logistic Regression, and Naïve Bayes Classifier. To evaluate the performance of each model, we used confusion matrix, ROC curve and reported accuracy, recall and precision.

The evaluation parameters (figures) are shown below.

<1> SVM Classifier (Form 16, Figure 33, 34)

SVM Classifier		
Confusion Matrix	$\begin{bmatrix} 61 & 29 \\ 4 & 42 \end{bmatrix}$	
Accuracy	Recall	Precision
0.757353	0.913043	0.591549

Form 16 – Evaluation parameters of SVM Classifier

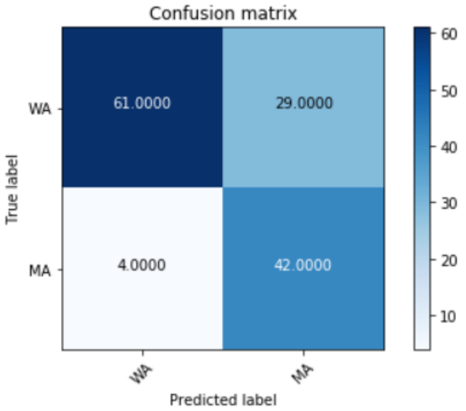

Figure 33 – Confusion Matrix of SVM Classifier

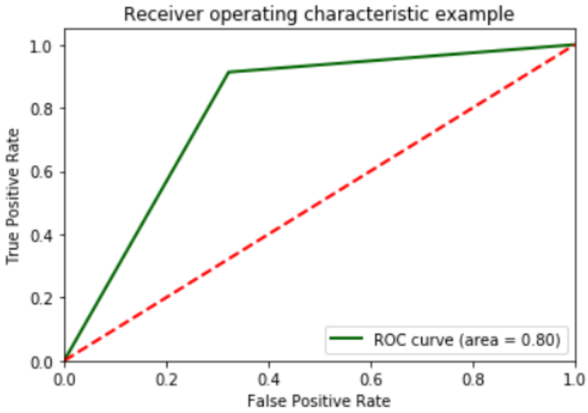

Figure 34 – ROC Curve of SVM Classifier

<2> Logistic Regression (Form 17, Figure 35, 36)

Logistic Regression Classifier		
Confusion Matrix	$\begin{bmatrix} 68 & 22 \\ 10 & 36 \end{bmatrix}$	
Accuracy	Recall	Precision
0.764706	0.782609	0.620690

Form 17 – Evaluation Parameters of Logistic Regression

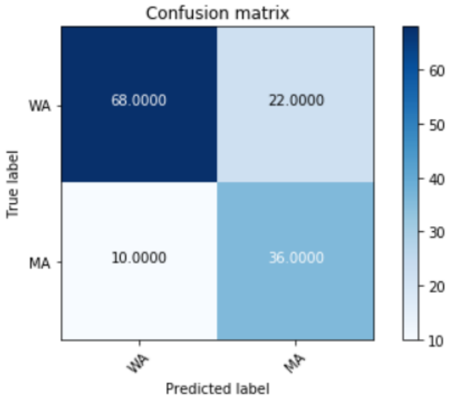

Figure 35 – Confusion Matrix of Logistic Regression

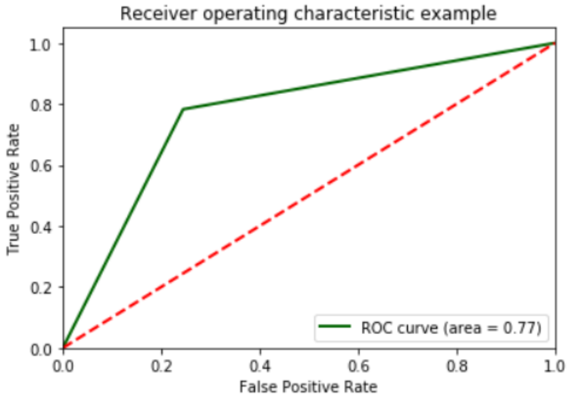

Figure 36 – ROC Curve of Logistic Regression

<3> Naïve Bayes Classifier (Form 18, Figure 37, 38)

Naïve Bayes Classifier		
Confusion Matrix	$\begin{bmatrix}62 & 28\\ 5 & 41\end{bmatrix}$	
Accuracy	Recall	Precision
0.757353	0.891304	0.594202

Form 18 – Evaluation Parameters of Naïve Bayes Classifier

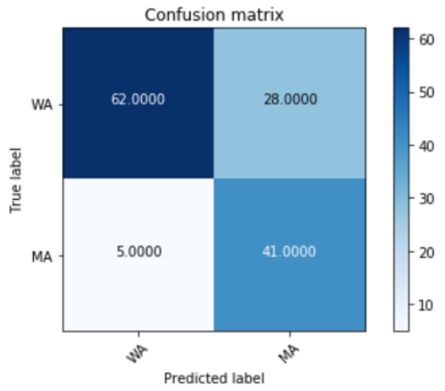

Figure 37 – Confusion Matrix of Naïve Bayes Classifier

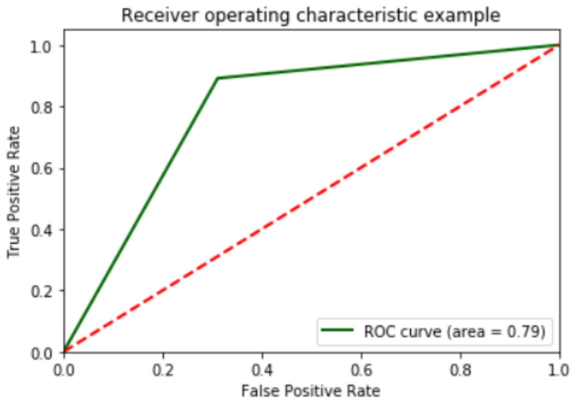

Figure 38 – ROC Curve of Naïve Bayes Classifier

From the ROC Curve and evaluation parameters, among the trained 3 types of classifiers, when applying SVM classifier, we could get a relatively better performance.

Part 3: Sentiment Analysis

Another concentration we've tried to pay on the twitter data is the "sentiment" analysis. For a national sports event like Superbowl, especially for the fans of two different teams, their moods could be impacted by the procedure or the result of the final, and reflected on the tweets they sent. Besides, in the different period, the mood change among the authors of tweets would vary.

In part 3, we designed our own model to carry out the sentiment analysis on twitter data. First, we loaded the data, obtaining the text of every tweet in files with different hashtags. Secondly, we used the function $SentimentIntensityAnalyzer()$ from $nltk$ library to get the information about these sentiments expressed in the text. We define 'positive' = 1, 'negative' = -1, and 'neutral' = 0. Finally, we calculate the sum of sentiment value per hour and plotted the figure for value versus time.

We analyzed the files with hashtag “#GoPatriots” and “#GoHawks”. The results are shown in Figure 39 and Figure 40.

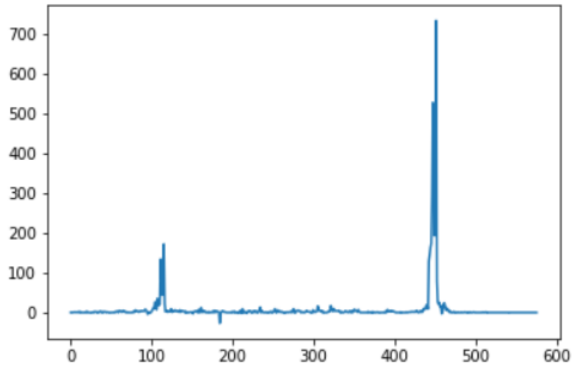

Figure 39 – Sentiment Analysis for #GoPatriots

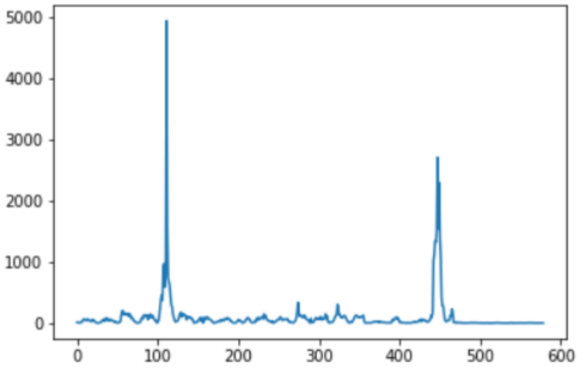

Figure 40 – Sentiment Analysis for #GoHawks

Since the data of each hashtag comes from two different datasets, the number of tweets contained in each dataset is distinctive, making the peak values have difference.

In realistic circumstance, Patriots won, and the plotted result could reflect this fact to some extent. First, the peak point of two figures shows the critical time, for instance, the second peak point is in accordance to the time of final game. In addition, a significance decreasing in the peak value of figure #GoHawks could be witnessed, and in contrast the peak value of figure #GoPatriots has a huge increase.

The sentiment value is calculated by taking all kinds of sentiments into account. For the results shown above, we proposed a prediction that the fans of winning team would show more positive emotions in their tweets, indicating a larger ratio of positive weight and a higher peak value, and vice versa.